In response to the Office Action mailed July 9, 2004, Applicant respectfully requests the consideration of the following remarks and, accordingly, the reconsideration of the Final Rejection of the claims. It is respectfully submitted that the pending claims are in condition for allowance.

REMARKS

Status of the Claims

Claims 1-15 were originally presented for examination. All of the original claims were rejected in the Office Action of July 9, 2004. Applicants have amended claims 1, 7, and 13-15, have canceled claims 8-11, and have added new Claim 16. The claims now pending are claims 1-7 and 12-16.

Status of Amendments

There are no unentered amendments.

Summary of the Invention

Applicants' claimed invention is a method, system, and program product for controlling access, in real time, to a file among a plurality of users. The first claimed step is establishing an object comprising distinguishable groups of data. Each group of data has associated access criteria. This access criteria controls access to the groups of data. Specifically, each group of data has an associated user privilege for identifying separate groups of information to which the user may have access to within the groups of data and for setting a user's ID. This includes defining which users are allowed to access the object and associated information and user privileges.

The system has a cache memory for storing user ID's; and a cache memory for storing user access criteria along with access application code that is responsive to (1) the user ID, (2) user access criteria associated with the groups of data contained within an object, and (3) predetermined privileges for allowing controlled access to individual groups of data contained within the object by an individual user according to the user's access privileges. These various memories are searched sequentially for the user ID, the access criteria, and the target data.

Thereafter the user is served with a redacted copy of the object, that is, the system sends an electronic object to the user that contains the groups of information to which the user has access and that excludes groups of information to which the user does not have access.

The Rejections

In the Office Action of July 9, 2004 all of the claims were rejected under 35 USC §102(e) as being anticipated by U.S. Patent 6,085,191 to Fisher et al. for System And Method For Providing Database Access Control In A Secure Distributed Network.

It is stated in the Office Action that Fisher et al. describes

1) an object and other associated information

2) distinguishable groups of data[1],

3) each group of data having associated access criteria for access to the groups of data and operation types[2];

---

[1] Referring also to FIG. 11A, the format of each row in the database tables preferably includes a field called the "fully distinguished name" (FDN) of a managed object followed by columns of data: Data 1, . . . , Data N. Preferably, the FDN for each row represents the tree path (through the managed object tree) for the managed object whose information is stored in that row. The tree path for an object may be represented in the form "/a/b/c/ . . . " where a, b, and c indicate nodes along the tree path. For example, an FDN can look like:

/systemid="sys1"/owner="abccompany"/devicetype="router"/ . . .

The FDN operates as the primary key to the data stored in the table. Using security mechanisms that will be described below, the FDN is used as the key that determines which managed objects that a particular user is permitted to access or modify.

Referring back to FIG. 10, unlike the embodiment described above, in this embodiment, the rows 311, 312, 321, 322 of the tables 310, 320 contain management information for managed objects associated with the networks of many users. The present invention provides a way to restrict access to the management information in the database tables 310, 320 so that each user is allowed access only to the management information that the user would be allowed to access by the access control server 330. That is, the same management information access rights and restrictions that are represented by the access control tree 170 are also enforced by the DBMS. (Fisher et al., Column 19, lines 1-29)

[2] Creating the Permission Tables

The system administrator 302 creates the permissions tables prior to use of the DBMS 280 by end users. A call 440 to a create permissions procedure 442 is used to create the permissions tables. The "Create.sub.-- Permissions.sub.-- Tables" procedure 442 is stored in the DBMS 280. The system administrator 302 invokes a call 440 to the "Create.sub.-- Permissions.sub.-- Tables" 442 procedure of the DBMS 280. The Create.sub.-- Permissions.sub.-- Tables 442 procedure maps the user access rights as defined by the access control tree 170 into the proper format for the permissions tables.

4) a central processing unit (CPU) for controlling the access to the database; a memory for storing software code for controlling the operation of the CPU[3];

5) access application code stored in the memory and executable by the CPU[4].

6) access control procedure checks both "Grant" and "Deny" tables.[5]

Referring also to FIGS. 15A and 15B, a permission entry 1502, 1504 is tuple having three fields, as shown below:

(user name, object name, operation type).

Although FIGS. 15A and 15B show the object name in each permission entry as a single "word," preferably the object name is the FDN for a managed object. The user name is the name of the user (or the group of users) whose access rights are represented by the permission entry, the object name identifies the managed object to which the permission entry applies, and the operation type is the operation that the specified user is being granted or denied with respect to the specified object. The operation type can be a select, delete, insert or update operation.

The two permissions tables reflect the manner in which the X.741 access rules are specified: Global grant, Global grant with item deny, Global deny with item grant, and Global deny.

The Grant table 408 stores a list of all access Grant permissions. The Deny table 410 stores a list of all access Deny permissions. When checking whether access should be permitted for a particular operation, the access control procedure 404 checks both tables. (Column 25, line 36 to column 26, line 2)

[3] The MIS 150 includes:

an interface 160 for receiving access requests;

one or more central processing units (CPU's) 162 for executing access control procedures stored in the MIS's memory 164;

memory 164, including both volatile high speed RAM and non-volatile storage such as magnetic disk storage;

an interface 166 for handling secure communications between the MIS 150 and the auxiliary access control servers 152, 154; and

one or more internal busses 168 for communicating data and programs between the above referenced elements of the MIS 150. (Column 7, line 25-38

[4] See footnote 3, column 7, lines 25-35.

[5] Although FIGS. 15A and 15B show the object name in each permission entry as a single "word," preferably the object name is the FDN for a managed object. The user name is the name of the user (or the group of users) whose access rights are represented by the permission entry, the object name identifies the managed object to which the permission entry applies, and the operation type is the operation that the specified user is being granted or denied with respect to the specified object. The operation type can be a select, delete, insert or update operation.

The two permissions tables reflect the manner in which the X.741 access rules are specified: Global grant, Global grant with item deny, Global deny with item grant, and Global deny.

The Grant table 408 stores a list of all access Grant permissions. The Deny table 410 stores a list of all access Deny permissions. When checking whether access should be permitted for a particular operation, the access control procedure 404 checks both tables.

Populating the Permissions Tables

The permissions tables 406 are populated to correspond to the access control rules 206 of the access control tree 170. By convention, the permissions tables 406 use a special object name value, such as a database NULL value, to represent "all objects," and a special operation type value, such as a database NULL value, to represent "all operation types."

The permissions tables 406 are populated as follows:

If the rule in the access control database specifies a "global grant" to user U1 for operation type Op1, an entry is made in the grant table 408 which is (U1, NULL, Op1).

If the rule in the access control database specifies "global grant to user U1 with item deny for items X1, X2 and X3" for operation type Op1, the following entries are made in the grant table 408 and the deny table 410:

GRANT TABLE:	(U1, NULL, Op1)
DENY TABLE:	(U1, X1, Op1)
	(U1, X2, Op1)
	(U1, X3, Op1)

If the rule in the access control database specifies "deny user U1 access to all items except items X1, X2 and X3" for operation type Op1, then the following entries are made in the deny table 410 and grant table 408:

DENY TABLE:	(U1, NULL Op1)
GRANT TABLE:	(U1, X1, Op1)
	(U1, X2 Op1)
	(U1, X3, Op1)

If the rule says "global deny" to user U1 for the operation type Op1, the following entry is made in the deny table 410:

DENY TABLE: (U1, NULL, Op1)

In all the above, multiple entries can be made for different operation types. If the user's permissions on the managed object are the same for all operation types, then a single entry with the NULL operations type will suffice.

The method described above is a more efficient way to store access control rules than storing only explicit grant rules or storing only explicit deny rules. For example, if one were to store only grant rules, then in a

7) the operations of reading, modifying, and deleting data with authorized access.

8) adding data[6]

9) the data tables show an object and associated information.[7]

10) where each row of the table has the FDN (Fully Distinguished Name) and the primary key[8]

12) checking the "Grant" and "Deny" tables for access permission for a particular operation[9]

---

system with 5,000 managed objects, a new user given a global grant with a single item deny would require 4,999 records in the Grant Table 408. Using the method described above, the new user would have just two entries: one entry in the Grant Table and another entry in the Deny table 410. (Column 25, line 51 to column 26, line 52)

[6] In addition to rule objects that specify a set of target managed objects, the system can have one global deny rule object and one global allow rule object. Each of the global rule objects has the same structure as a regular rule object, but has any empty target list field, which indicates the rule is a global rule. The global deny rule, if defined, specifies groups of users that cannot perform any operations on any managed objects. The global grant rule, if defined, specifies groups of "super users" (e.g., system administrators) that are allowed to perform all operations on all managed objects. (Column 11, lines 8-17)

[7] Referring to FIG. 10, a direct information access system 1000 that uses views is shown. The primary components of the direct information access system 1000 are: a conventional DBMS 280 for storing tables 310, 320, an access control server 330, an access control tree 170, an information transfer module 340, and a network 106. The network 106 and access control tree 170 have been described above. (Column 18, lines 39-42)

[8] Referring also to FIG. 11A, the format of each row in the database tables preferably includes a field called the "fully distinguished name" (FDN) of a managed object followed by columns of data: Data 1, . . . , Data N. Preferably, the FDN for each row represents the tree path (through the managed object tree) for the managed object whose information is stored in that row. The tree path for an object may be represented in the form "/a/b/c/ . . . " where a, b, and c indicate nodes along the tree path. For example, an FDN can look like:

/systemid="sys1"/owner="abccompany"/devicetype="router"/ . . .

The FDN operates as the primary key to the data stored in the table. Using security mechanisms that will be described below, the FDN is used as the key that determines which managed objects that a particular user is permitted to access or modify.

Referring back to FIG. 10, unlike the embodiment described above, in this embodiment, the rows 311, 312, 321, 322 of the tables 310, 320 contain management information for managed objects associated with the networks of many users. The present invention provides a way to restrict access to the management information in the database tables 310, 320 so that each user is allowed access only to the management information that the user would be allowed to access by the access control server 330. That is, the same management information access rights and restrictions that are represented by the access control tree 170 are also enforced by the DBMS. (column 19, lines 1-29)

13) global grants of access by a global entry in the grant table.[10]

14) the sequence of receiving an object request by a user, verifying the user privilege access criteria, and transmitting information according to the user's user privilege access criteria.[11]

---

9 The Grant table 408 stores a list of all access Grant permissions. The Deny table 410 stores a list of all access Deny permissions. When checking whether access should be permitted for a particular operation, the access control procedure 404 checks both tables. (Column 25, line 65 to column 26, line 2)

10 See footnote 5.

11 Enforcing Access Control

Enforcement of Access Control Rules based on permission tables is done according to the following algorithm, which assumes that an operation is requested by user U1.

These steps are followed in sequence unless a grant or deny decision is reached in any one step, in which case the algorithm exits.

1. Check the Deny table to see if the User U1 has a global deny (i.e., a deny to all objects). If so, check the Grant table to see if the user has specific granted items (objects) that are exceptions to the deny. If any such objects exist, grant access if the current operation matches the operation specified in the Grant table, otherwise deny access.

2. Check the Grant table to see if the User U1 has a global grant (i.e. a grant to all objects). If so, check the Deny table to see if the user has specific denied items (objects) that are exceptions to the grant. If any such objects exist, deny access if the current operation matches the operation specified in the Deny table, otherwise grant access.

3. Check the Deny table to see if User U1 has specific denied items (objects), and deny access if the current operation matches the operation specified in the Deny table.

4. Check the Grant table to see if User U1 has specific granted items, and grant access if the current operation matches the operation specified in the Grant table.

5. Check the Deny table to see if there is an all-users global deny (i.e., deny all objects to all users). If so, check the Grant table to see if all users have specific granted items (objects) that are exceptions to the deny. If any such objects exist, grant access if the current operation matches the operation specified in the Grant table, otherwise deny access.

6. Check the Grant table to see if there is an all-users global grant (i.e., grant all objects to all users). If so, check the Deny table to see if all users have specific denied items (objects) that are exceptions to the grant. If any such objects exist, deny access if the current operation matches the operation specified in the Deny table, otherwise grant access.

7. Check the Deny table to see if all users have specific denied items (objects). If so, deny access if the current operation matches the operation specified in the Deny table.

8. Check the Grant table to see if all users have specific granted items (objects). Grant access if the current operation matches the operation specified in the Grant table.

9. If no grant/deny decision has been reached following the steps above, apply the default Access Control Rule (default deny or default grant).

15) limiting views to certain columns (to which the user does not have access) in a document or file to which the user otherwise has access.[12] Note that this is serving a copy of the redacted document to the user, simply displaying portions of the document.

It is conceded in the Office Action that Fisher et al. does not describe Applicants' claimed features of:

1) the application code being responsive to the access criteria associated with the groups of data contained within an object and to

---

When the user submits a query requesting access to multiple objects, such as a request for the status of all routers in the network or a request for information about a specified list of managed objects, the access control procedure 404 performs the applicable access rights checking method for all the requested objects. Access is allowed only for the objects to which the user has appropriate access rights. No information is returned to the user for other objects in the database, and thus the user is not informed that access has been denied to any objects. This is important, because the user must not be informed of even the existence of objects that are not within his purview. Also, the user should be able to simply request information about "all objects" of a particular type, without having to be concerned about excluding objects from the query to which the user does not have access.

If access to all the objects specified in a query is denied, the query is denied without providing a detailed explanation to the user. If access is granted for some object but not others, the access control procedure 404 enables the user query to be executed on the objects for which access is granted. In particular, in the preferred embodiment the access control procedure 404 executes the user query against those objects and returns the results to the user. As a result, the normal query processing by the database access engine is circumvented and replaced by processing performed by (or initiated by) the access control procedure 404. The data read from the DBMS tables by the access control procedure 404 is returned to the requesting user or process in the same way that the data would have been returned if the query had been processed by the database access engine.

(Column 26, line 61 – column 28, line 12)

12 To limit user access to the management information stored in the tables, this second embodiment uses a database function called "Views." The database access engine 286 of the DBMS 280 has a module that implements Views. Views are well known tools used by database engines. Views are sometimes used to make it easier for unskilled users to generate queries, for instance by assigning easy to remember aliases to database table columns. It is also well known that Views can also be used to limit the columns and rows of database tables that are accessible to users.

A View can be used to limit access by, in essence, hiding certain columns and rows from the user, or alternately by limiting the user's access to specifically designated columns and rows. Some database engines also provide a security mechanism for limiting use of a particular View to a specified set of users. These functions of Views are well known to those skilled in the art and will not be described in detail, except to the extent that they are utilized by this embodiment of the present invention. (Column 19, lines 30-49)

2) the application code being responsive to predetermined privileges for allowing controlled access to individual groups of data contained within the object by an individual user according to the user's privileges.

3) transmitting a redacted object including sending an electronic object to the user where the object itself (and not just an on-screen display thereof) contains the groups of information to which the user has access to and excludes information to which the user does not have access.

The Reference

The sole reference relied upon is United States Patent 6,085,191 to Fisher et al. for System And Method For Providing Database Access Control In A Secure Distributed Network. Fisher et al. describe an access control database that has access control objects. These access control objects collectively store information that specifies access rights of users to specified sets of the managed data objects. The specified access rights include access rights to obtain information from the network.

As described in Fisher et al. an access control server provides users access to the managed objects in accordance. This is in accordance with the access rights specified by the access control database. An information transfer mechanism sends management information from the network to a database management system (DBMS) for storage in a set of database tables.

Each database table stores management information for a corresponding class of managed objects. A set of views limits access to the management information stored in the database tables. Each view defines a subset of rows in the database tables that are accessible when using this view.

The set of database table rows that are accessible when using each view in the set correspond to the managed object access rights specified by the access control database. A view access control means specifies which views in the set of views are useable by specified ones of the users. The database access engine accesses information in the set of database tables using the set of views such that each user is allowed access only to management information in the set of database tables that the user would be allowed by the access control database to access.

Issue

The sole issue presented is whether the pending claims are properly allowable to Applicants over United States Patent 6,085,191 to Fisher et al. More particularly, the issue is whether Fisher et al.'s disclosure of standard database access control and system access control features[13] where the sole reference, Fisher et al., fails to show the claimed features as:

1) The application code is claimed as being responsive to the access criteria associated with the groups of data contained within an object and to

2) The application code is claimed as being responsive to predetermined privileges for allowing controlled access to individual groups of data contained within the object by an individual user according to the user's privileges.

3) The claims recite transmitting a redacted object including sending an electronic object to the user that contains the groups of information to which the user has access to and excludes information to which the user does not have access. Sending the redacted object itself is clearly distinguishable from merely providing an on-screen display.

---

[13] 1) Objects and other associated information

2) Distinguishable groups of data.

3) Where each group of data has associated access criteria for access and operation types.

4) Standard system features such as a central processing unit (CPU) for controlling the access to the database; and a memory for storing software code for controlling the operation of the CPU.

5) Access application code stored in the memory and executable by the CPU.

6) An access control procedure that checks both "Grant" and "Deny" tables for access permission for a particular operation

7) Once access has been granted performing the operations of reading, modifying, adding, and deleting data.

8) Global grants of access by a global entry in the grant table.

9) The sequence of receiving an object request by a requester, verifying the user privilege access criteria, and transmitting information according to the requestor's user privilege access criteria.

10) Limiting views to certain columns (to which the requestor does not have access) in a document or file to which the requestor otherwise has access as a form of redacting.

Argument

Summary of The Argument

As conceded in the Office Action, fails to teach critical aspects of Applicants' claimed invention, i.e.,

1) The application code being responsive to the access criteria associated with the groups of data contained within an object and to

2) The application code being responsive to predetermined privileges for allowing controlled access to individual groups of data contained within the object by an individual user according to the user's privileges.

3) Transmitting a redacted object including sending an electronic object to the user that contains the groups of information to which the user has access to and excludes information to which the user does not have access which is clearly distinguishable from merely displaying portions of the object.

Applicants' claimed invention is a method, system, and program product for controlling access, in real time, to a file among a plurality of users. The first claimed step is establishing an object comprising distinguishable groups of data. Each group of data has associated access criteria. This access criteria controls access to the groups of data. Specifically, each group of data has an associated user privilege for identifying separate groups of information to which the user may have access to within the groups of data and for setting a user's ID. This includes defining which users are allowed to access the object and associated information and user privileges.

The system has a cache memory for storing user ID's; and a cache memory for storing user access criteria along with access application code that is responsive to (1) the user ID, (2) user access criteria associated with the groups of data contained within an object, and (3) predetermined privileges for allowing controlled access to individual groups of data contained within the object by an individual user according to the user's access privileges.

Moreover, the method and program product claims positive recite extracting the user's user identification from the object request, verifying (first in cache memory and if not in cache then in main memory) the user's user identification and identifying the groups of data to which the user has access and privileges with respect thereto. This allows controlled access to individual groups of data contained within the object by an individual user according to the user's privileges. Next the claims recite searching for the data first in cache and if not found in cache then in main memory and retrieving the data requested according to the user's access criteria.

The claimed invention further claims transmitting a redacted object to the user, i.e., sending an electronic object to the user that contains the groups of information to which the user has access to and that excludes groups of information to which the user does not have access.

Objections Under 35 USC§112(2)

Claims 9-12 have been rejected under 35 USC §112. These claims have been canceled, and the limitations (objected to as lacking antecedent basis) have been placed in the base claim.

CONCLUSION

Claims 1-7 and 12-16 are pending. Claims 1, 7, and 13-15 have been amended. Claim 16 has been added. Claims 8-11 have been cancelled. Applicants respectfully submits that, in view of the discussion set forth herein, the pending claims are patentable over the prior art.

The Commissioner is hereby authorized to charge any additional fees due or credit any overpayment to Deposit Account No. 50-2421.

If there are any questions regarding this correspondence, please contact the undersigned at (408) 288-7588.

Respectfully Submitted,

Dated: October 12, 2004 By:

David R. Stevens
Reg. No. 38,626